

APPLICATION FOR UNITED STATES LETTERS PATENT

TITLE:	AUTOMATED BANKING MACHINE COMPONENT AUTHENTICATION SYSTEM AND METHOD
INVENTORS:	PARSONS, ET AL.
DOCKET:	D-1170 R

## CROSS REFERENCE TO RELATED APPLICATIONS

This application claims benefit of U.S. Provisional Application Serial Nos. 60/396,608 filed July 16, 2002 and 60/436,883 filed December 26, 2002.

## TECHNICAL FIELD

5           This invention relates to automated banking machines. Specifically this invention relates to an automated banking machine in which individual components of the machine are operative to authenticate each other.

## BACKGROUND ART

10           Automated banking machines are well known. A common type of automated banking machine used by consumers is an automated teller machine ("ATM"). ATMs enable customers to carry out banking transactions. Common banking transactions that may be carried out with ATMs include the dispensing of cash, the making of deposits, the transfer of funds between accounts, the payment of bills and account balance inquiries. The types of banking transactions a customer can carry out are determined by capabilities of the particular banking machine and the programming of the institution operating the machine. Other types of automated banking machines may allow customers to charge against accounts or to transfer funds. Other types of automated banking machines may print or dispense items of value such as coupons, tickets, wagering slips, vouchers, checks, food stamps, money orders, scrip or traveler's checks. For purposes of this disclosure an ATM, an automated banking machine, or an automated transaction machine shall encompass any device which carries out transactions including transfers of value.

15

20

ATMs may include a plurality of devices including for example a CPU processor board, a hard drive, a keypad, and cash dispenser. ATM cases may have a plurality of designs and shapes. For example, ATMs may include a large reinforced security chest or safe which is capable of enclosing both a cash dispenser mechanism and a computer which operates the cash dispenser as well as the other devices of the ATM. In other ATMs, the computer of the ATM may be located outside the chest, although still within a locked enclosure or fascia. Unfortunately, an enclosure or fascia may be less secure than a chest and may be pried or cracked open. As a result computers or other ATM devices located outside the chest may have an increased risk of being modified or hacked by unauthorized users. Such modifications may compromise the security of the ATM and improperly cause the ATM to dispense cash or otherwise transfer value to the unauthorized user. Consequently there exists a need for an automated banking machine that has increased protection against unauthorized access to physical hardware devices of the machine.

In addition, ATMs are connected to at least one network. Such networks may include private networks which include one or more ATM host banking systems. Such networks may also include public networks such as the Internet. ATMs may also use standard Internet protocols to communicate with ATM host banking systems. Such standard protocols may include network protocols such as TCP/IP. As a result ATMs which use TCP/IP may be attacked with the same types of hacking tools used to attack web sites, and other types of computer systems on the Internet. Consequently there exists a need for an automated banking machine that has increased protection against unauthorized access to the machine through network communication.

Once an unauthorized user has gained access to the hardware of an ATM, whether by network communication or physical access to the hardware, the unauthorized user may have the opportunity to steal information from the ATM. Such modifications may include the insertion of viruses, worms, and/or sniffer programs which are operative to capture transaction information such as account numbers, personal identification numbers, and other secret information. As a result there further exists a need for an automated banking machine which has increased protection against the theft of transaction information.

### DISCLOSURE OF INVENTION

It is an object of an exemplary form of the present invention to provide an automated banking machine at which a user may conduct transactions.

It is a further object of an exemplary form of the present invention to provide an automated banking machine which is more secure.

It is a further object of an exemplary form of the present invention to provide an automated banking machine that has increased resistance to being attacked by an unauthorized user.

It is a further object of an exemplary form of the present invention to provide an automated banking machine which is operative to prevent unauthorized modifications to the ATM from enabling the theft of cash or transaction information.

Further objects of exemplary forms of the present invention will be made apparent in the following Best Modes for Carrying Out Invention and the appended claims.

The foregoing objects may be accomplished in an exemplary embodiment by an automated banking machine that includes output devices such as a display screen and receipt printer. The machine may further include input devices such as a touch screen, keyboard, keypad, function keys, and a card reader. The automated banking machine may further include transaction function devices such as a cash dispenser mechanism for sheets of currency, a depository mechanism and other transaction function devices which are used by the machine in carrying out banking transactions including transfers of value. In the exemplary embodiment the automated banking machine may include at least one computer. The computer may be in operative connection with the output devices and the input devices, as well as with the cash dispenser mechanism, depository mechanism and other physical transaction function devices in the banking machine. The computer may further be operative to communicate with a host system located remotely from the machine.

In the exemplary embodiment, the computer may include software components that are executable therein. The software components of the automated banking machine may be operative to cause the computer to output user interface screens through a display device of the machine. The user interface screens may include consumer screens which provide a consumer with information for performing consumer operations such as banking functions with the machine. The user interface screens may further include service screens which provide an authorized user servicing the machine with information for performing service and maintenance operations with the machine. In addition the machine may include software components operative in the computer for controlling and communicating with hardware devices of the machine including the input devices, output devices and the transaction function devices.

In an exemplary embodiment the automated banking machine includes a plurality of components which may correspond to two or more of the previously described input devices, output devices, transaction function devices, and/or software components. The plurality of components may include a first component and a second component. The first and second components are operative to securely communicate with each other. Such secure communications may include passing encrypted information between each other which is associated with a transaction function being performed by the machine. Such a transaction function may correspond to the dispense of cash or any other function performed by the automated banking machine. The information may be encrypted with a secret key such as a DES key known to both the first and second components. The encrypted information may correspond to account numbers, personal identification numbers (PINs), status information, error information, encryption keys, software function or event calls, operational commands, or any other type of information which may be passed between hardware and/or software components of an automated banking machine for purposes of performing a transaction function.

When the secure communication between the first and second components is being established, the second component of the exemplary embodiment may be operative to authenticate the identity of the first component in order to prevent the possibility of a "man in the middle" hacking attack. Such a "man in the middle" hacking attack may correspond to a software virus, worm, or an imposter hardware device or software component which is constructed to intercept secure communications between components. In one scenario of "a man in the middle" attack the rogue software or hardware component may function as a router which filters the intended communications between the first and second components. Such filtering may enable

the rogue component to acquire the secret key or keys used to decrypt the information being communicated between the first and second components. Such filtering may also enable a rogue component to replay sampled communications to cause a hardware device or other component to repeat a function.

5           The exemplary embodiment of the present invention may be operative to prevent "a man in the middle" attack by requiring the first component to authenticate itself to the second component with first identity data which is not generally available to a rogue software or hardware component. Such first identity data may correspond to individual or combinations of unique data embedded within the component. For example such first identity data may  
10 correspond to a serial number stored in a BIOS of a hardware device or may be embedded in the software code of a software component.

In the exemplary embodiment, the first component may be operative to randomly generate the secret key which will be used to perform symmetric encryption/decryption of information being communicated between the first and second components. To facilitate the  
15 transfer of the secret key to the second component, the second component may include a public and private key set which may be used to perform asymmetric encryption/decryption. In the exemplary embodiment, the first component may be operative to encrypt the secret key with the public key of the second component. The encrypted secret key may then be sent to the second component. The second component may be operative to decrypt the secret key using its private  
20 key.

To enable the second component to verify that the encrypted secret key originated from an authentic first component, the first component may be operative to generate at least one first

hash using at least one identity data values embedded in or accessible to the first component.

The first hash may be generated by hashing the first identity data with the public key using one-way hash algorithms such as MD5 and SHA-1.

Both the encrypted secret key and the first hash of the first identity data may be sent from the first component to the second component as part of at least one message. In an exemplary embodiment, the first component may further encrypt the first hash with the secret key before sending it to the second component. In other exemplary embodiments, the first component may encrypt a combination of the secret key and first hash with the public key before sending it to the second component.

The second component is operative to decrypt the secret key with the private key of the second component. In embodiments where the first hash is encrypted with the secret key, the second component is further operative to decrypt the first hash using the secret key. In embodiments where a combination of the secret key and the first hash are encrypted with the public key, the second component may be operative to decrypt the combination using the private key to uncover both the secret key and the first hash.

To verify that the secret key originated from an authentic first component and not an imposter first component, the second component may be operative to compare the first hash to a second hash that is expected to correspond to the first component. In an exemplary embodiment, the second component may access a secure data store which includes at least one second identity data which corresponds to the first component. The second component may be operative to cause at least one second hash to be generated by passing the second identity data and the public



key through a one-way hash function which is identical to the function used by the first component to generate the first hash.

If the first and second hashes correspond to each other, the second component is operative to permit secure communications between the first and second components to commence using the secret key to encrypt the communications. If the first and second hashes do not correspond to each other, the second component is operative to not establish secure communications between the first and second components using the secret key received from the first component. The second component may also propagate an error message in the automated banking machine which indicates that the first component may not be an authorized first component.

In exemplary embodiments, the described cryptographic processes may be performed by the software components. In addition, in exemplary embodiments the described cryptographic processes may be performed by cryptographic software or hardware module of the automated banking machine responsive to the components. For example an exemplary embodiment of the automated banking machine may include a message protect application which is operative to generate the first and second hashes. In an exemplary embodiment each of the first and second component may call a separate instance of the message protect application. For example, if each of the first and second components include separate processors, each processor may run a version of the message protect application.

In an alternative exemplary embodiment, the ATM may include a trusted platform with a trusted platform module (TPM) and associated software which complies with a trusted computing platform or base specification. The trusted platform may be used to perform all or

portions of the cryptographic functions for the software and hardware components of the automated banking machine to establish secure communications between the components.

### BRIEF DESCRIPTION OF DRAWINGS

Figure 1 is a perspective view representative of an exemplary embodiment of an automated banking machine.

Figure 2 is a schematic view of a further exemplary embodiment of an automated banking machine.

Figure 3 is a schematic view showing a method for establishing secure communications between components of an automated banking machine.

Figure 4 is a schematic view showing secure communications between terminal control software components and a cash dispenser and a card reader of an automated banking machine.

Figure 5 show an exemplary embodiment of an automated banking machine with a software component of a computer that is operative to establish a secure communication session with a cash dispenser.

Figure 6 show an schematic view of an automated banking machine with a software component of a computer that is operative to establish a secure communication session with a cash dispenser.

Figures 7 and 8 show schematic views of an automated banking machine that includes a trusted platform with a trusted platform module.

### BEST MODES FOR CARRYING OUT INVENTION

Referring now to the drawings and particularly to Figure 1, there is shown therein a perspective view of an exemplary embodiment of an automated banking machine 10. Here the automated banking machine 10 may include at least one output device 34 such as a display device 12. The output device 12 may be operative to provide a consumer with a user interface 18 that may include a plurality of screens or other outputs including selectable options for operating the machine. The exemplary embodiment may further include other types of output devices such as a receipt printer 20, statement printer 21, speakers, or any other type of device that is capable of outputting visual, audible, or other sensory perceptible information.

The exemplary embodiment of the automated banking machine 10 may include a plurality of input devices 32 such as an encrypting pin pad (EPP) with keypad 16 and function keys 14 as well as a card reader 22 and/or bar code reader 23. The exemplary embodiment of the machine 10 may further include or use other types of input devices, such as a touch screen, microphone, or any other device that is operative to provide the machine with inputs representative of user instructions or information. The machine may also include one or more biometric input devices such as a fingerprint scanner, an iris scanner, facial recognition device, hand scanner, or any other biometric reading device which may be used to read a biometric input that can be used to identify a user.

The exemplary embodiment of the automated banking machine 10 may further include a plurality of transaction function devices which may include for example a cash dispenser 24, a depository mechanism 26, a cash recycler mechanism, or any other type of device which is operative to perform transaction functions involving transfers of value.

Figure 2 shows a schematic view of components which may be included or may be in communication with the automated banking machine 10. Exemplary embodiments of the automated banking machine 10 may be operative to communicate with a transaction processing server which is referred to herein as an ATM host banking system 42. Such an ATM host banking system 42 may be operative to authorize the automated banking machine 10 to perform transaction functions for users such as withdrawing cash from an account through operation of the cash dispenser 24, depositing checks or other items with the depository mechanism 26, performing a balance inquiry for a financial account and transferring value between accounts.

In addition, the machine 10 may include at least one computer 30. The computer 30 may be in operative connection with a plurality of components 44. Such components may include both hardware components 46 and software components 40. The hardware components 46 may correspond to the previously described input device(s) 32, output device(s) 34, and transaction function device(s) 36. In an exemplary embodiment, a transaction function device may be operative to perform a transaction function in response to at least one input through at least one of the input devices.

In an exemplary embodiment, the software components may correspond to one or more terminal control software components that are operative in the computer 30. The terminal control software components may be operative to control the operation of the machine by both a consumer and an authorized user such as a service technician. For example such terminal control software components may include applications which enable a consumer to dispense cash, deposit a check, or perform other transaction functions with the machine. In addition the

terminal control software components may include applications which enable a service technician to perform configuration, maintenance, and diagnostic functions with the machine.

In an exemplary embodiment both software components and/or hardware components may be operative to communicate with each other through the computer and/or through a communication system associated with the computer. Such communications may be encrypted to prevent hacking attacks in which rogue software and/or hardware components access the communications. Figure 3 shows a schematic view of an exemplary method for establishing secure communications between a plurality of components of the automated banking machine 10. Here the automated banking machine may include at least two components such as a first component 102 and a second component 104. In the exemplary embodiment the first and second components may correspond to software components that are operative in the at least one computer processor of the automated banking machine. In addition in exemplary embodiments either or both of the first and second components may correspond to hardware components. As used herein, it is to be understood that the term component may correspond to either a hardware or software component of the automated banking machine.

The automated banking machine may also include one or more transaction function devices. Such a transaction function device may correspond to one or both of the first or second components 102 and 104. However, a transaction function device may also correspond to a further component in addition to the described first and second components 102, 104.

The first component 102 may be operative to randomly generate a secret key 112. The first component 102 may be operative to send at least one first message 114 to the second component which includes the secret key in an encrypted form 130. Subsequent messages 116

communicated between the first and second components may include information 118 which is encrypted with the secret key. In the exemplary embodiment the subsequent messages 116 may be encrypted and decrypted by the first and second components using a symmetrical cryptography algorithm and the secret key. An example of a symmetrical algorithm which may be used to encrypt the messages 116 may include the DES cryptography algorithm.

In the exemplary embodiment, the secret key may be randomly generated by the first component each time the first component and second component attempt to establish a secure and encrypted channel of communication between each other. In addition, to prevent the secret key from being uncovered from the first message 114, the exemplary embodiment of the first component may be operative to encrypt the secret key using an asymmetric cryptography algorithm and a public key associated with the second component. The asymmetric cryptography algorithm may include RSA or other public/private key cryptography algorithm. The resulting encrypted secret key 130 in the first message 114, may be decrypted by the second component using the asymmetric cryptography algorithm and a private key 122 of the second component. The first component 102 may be operative to acquire the public key 120 of the second component from a message 113 sent from the second component. In alternative exemplary embodiments, the first component 102 may be operative to acquire the public key 120 from a data store of public keys.

In the exemplary embodiment, the second component may be operative to verify that the secret key received in the message 114 originated from the first component. To enable such verification, the first component 102 may include a first identity data 106. Such first identity data may be embedded in at least one secure storage location within the component or may be

accessed by the component from associated software and/or hardware devices. For example, in a hardware component, the first identity data may correspond to a serial number or sub-byte identification codes stored in a BIOS or other non-volatile memory device in the hardware component. In a software component, the first identity data may correspond to a unique identification number stored in the software code of the software component. In addition, in exemplary embodiments, the identity data may be comprised of a combination of different portions of information securely stored in the component or accessed by the component from different components.

In the exemplary embodiment, the first component is operative to generate at least one first authentication hash of the at least one first identity data. Such a first hash may be generated by hashing both the first identity data and the public key of the second component that was used to encrypt the secret key. The term hash as used herein corresponds to a non-reversible value generated by passing data such as a combination of the identity data and the public key through a one-way hash function. Examples of a one-way hash function for use with exemplary embodiments include MD5 and SHA-1 or any other function which is operative to generate a non-reversible hash value from a combination of the first identity data and the public key.

In the exemplary embodiment, the first hash may be encrypted. The resulting encrypted first hash 132 may be sent to the second component in the at least one first message 114. In an exemplary embodiment, the first hash may be encrypted with the secret key 112 generated by the first component. In an alternative exemplary embodiment, the first hash may be combined with the secret key and the resulting combination may be encrypted with the public key 120 of the second component.

When the at least one first message 114 is received, the second component may be operative to decrypt the encrypted first hash 132 to reveal the first hash 108. In exemplary embodiments where the first hash is encrypted using the secret key, the second component may decrypt the first hash using the secret key received from the first component. In exemplary  
5       embodiments, where the first hash and secret key are encrypted together as a combination using the public key 120, the second component may be operative to decrypt the combination using the private key 122 of the second component.

In the exemplary embodiment, the second component may be operative to access at least one second authentication hash 134 which is associated with the first component 102. The  
10       second hash 134 may correspond to the hashing of at least one second identity data 136 and the public key 120 using the same hash function as was used to generate the first hash 108. The automated banking machine 10 may be configured such that the second identity data 136 is identical to the first identity data 106 if the first component is authentic and has been properly configured in the machine. As a result the first hash 108 and the second hash 134 will match  
15       when the first message 114 is from an authentic first component. The exemplary embodiment of the second component may be operative to compare the first hash to the second hash to verify that they are identical. If the first and second hashes are identical the second component may be operative to enable secure communications or messages 116 to be sent between the first and second components which include information 118 encrypted with the secret key 112. If the first  
20       and second hashes are not the same, the second component may be operative to propagate a status or error message in the automated banking machine which indicates that the second



component failed to establish a secure communication session or channel with the first component.

In the exemplary embodiment, the second component 104 may be operative to access the second identity data 136 and generate the second hash 134 from the second identity data 136 and its public key 120. In exemplary embodiment the second identity data 136 may be accessed from a data store 110 that is in operative connection with the second component. Such a data store 110 may be operative to securely store the second identity data such that an unauthorized rogue component cannot access the second identity data from the data store.

In an exemplary embodiment the data store 110 may be included in the second component 104. For example, if the component corresponds to a hardware device the data store 110 may correspond to a flash RAM or other nonvolatile memory of the hardware device.

In other exemplary embodiments the data store may correspond to a data store component that is accessible to a plurality of components and includes identity data for a plurality of components.

Components may securely access the data store component and retrieve identity data related to other components that are being authenticated. Such a data store component may itself correspond to the described first and/or second components and be operative to authenticate other components prior to enabling the components to access the identity data.

In an alternative exemplary embodiment, in which the computer of the ATM may include a trusted platform module (TPM) chip, the identity data may be stored in a secure data store managed by a trusted platform that uses TPM chip of the computer. Such a trusted platform may comply with a trusted platform specification such as such as the Trusted Computing Platform Alliance (TCPA) specification, the Microsoft next-generation secure computing base (NGSCB)

(formerly called Palladium) or other specification for establishing a trusted and secure computing environment.

Also, in a further exemplary embodiment, rather than having the second component generate the second hash 122, the secure data store 110 and/or associated software may be operative to generate the second hash 122 and securely send the second hash 122 to the second component. In this described exemplary embodiment, the data store 110 may receive or have access to the public key of the second component and may pass both the second identity data 136 and the public key 120 of the second component through a one-way hash function to generate the second hash 134. To prevent a rogue component from intercepting the second hash 134, the data store 110 may further encrypt the second hash 134 using the public key 120 of the second component. The second component may then decrypt the encrypted second hash 134 using its private key 122.

In the exemplary embodiment, the automated banking machine may include a user interface and/or a dedicated input device that is operative to enable an authorized user to maintain or update the identity data stored in the data store 110. When new or updated hardware or software components are installed in the automated banking machine 10, the data base 110 may be updated to include the new identity data values which correspond to the new components. As will be discussed in more detail below, such a dedicated input device may include a button accessible only with a secure area of the ATM such as the chest of the ATM. The button may be configured to cause the data store to be updated with new identity data for components currently installed in the ATM.

In alternative exemplary embodiments, the data store 110 or other data stores in the machine may include copies of the public keys for the components. Rather than having the components receive public keys from other components, the components may retrieve public keys for other components from a data store of public keys. In such alternative exemplary  
5 embodiment, data stores which include identity data and/or public keys may be operative to securely communicate with components to prevent rogue components from performing "man in the middle" attacks between the components and the data stores.

In exemplary embodiments, the components may include or have access to applications which provide cryptographic functions for performing, encryption, decryption, digital signature  
10 signing, digital signature verification, hashing and/or other cryptographic calculations as described herein. For example, as will be discussed in more detail below, the components may include or have access to a message protect application which is callable by the components and is operative to perform all or portions of the described steps for establishing a secure communication session between components.

15 In alternative exemplary embodiments, the automated banking machine may include dedicated software and hardware components which are operative to perform general cryptographic calculations on behalf of the calling components and/or a message protect application. For example, an ATM may include a dedicated hardware card or chip for performing cryptographic calculations such as random number generation and DES  
20 encryption/decryption. For computers of the ATM that include a TPM chip, the TPM and associated software may be operative to perform or manage cryptographic calculations responsive to software and hardware components of the automated banking machine.

Figure 4 shows an example of using the previously described system for authenticating components to perform a transaction function. Here the automated banking machine 10 includes a card reader 204, a cash dispenser 202, and terminal control software 200. The exemplary embodiment of the terminal control software 200 may include one or more software components such as a device driver which corresponds to at least one of the first components previously described. The exemplary embodiments of the card reader 204 and cash dispenser 202 may correspond to the second components previously described.

To perform a transaction function such as the dispensing of cash, the terminal control software components may require account information to be read from a card using the card reader 204. To prevent a virus, worm or other rogue component from intercepting the account information and to prevent a "man in the middle" attack, the previously described method of establishing a secure channel of communication may be used. In this example the card reader may be operative to encrypt the account information 206 being sent to the terminal control software using a secret key generated by the terminal control software. In addition the card reader may verify that the secret key was generated by an authentic terminal control software component 200, by comparing at least one first authentication hash to at least one second authentication hash as discussed previously. In this example the first and second hashes may be generated by passing a public key of the card reader and at least one identity data associated with or accessed by the terminal control software component through a one-way hash function.

To cause the automated banking machine to dispense cash, the terminal control software components 200 may pass a command to the cash dispenser device which indicates an amount to dispense. To prevent a virus, worm, foreign hardware device, or other rogue component from

being able to control the cash dispenser, the previously described method of establishing a secure communication session between the terminal control software and the cash dispenser may be used. In this example the terminal control software may be operative to encrypt the command instruction 208 to dispense an amount of cash using a secret key generated by the terminal control software. In addition the cash dispenser may verify that the secret key was generated by an authentic terminal control software component 200, by comparing at least one first authentication hash to at least one second authentication hash as discussed previously. In this example the first and second hashes may be generated by passing a public key of the cash dispenser and at least one identity data associated with or accessed by the terminal control software component through a one-way hash function.

In this described exemplary embodiment, both the card reader 204 and the cash dispenser 202 are operative to establish a secure communication session with one or more terminal control software components 200. In other exemplary embodiments, other hardware devices and/or software components of the machine may be operative to establish secure communication sessions between each other using the previously described methods. Also, individual components of the machine may correspond to a combination of the previously described first and second components and thus be able to authenticate other components as well as being able to be authenticated by other components as previously described.

In the previously described exemplary embodiment, a single identity data value such as a serial number may be used to authenticate a component. However, in alternative exemplary embodiments, a component may be authenticated using more than one identity data value. When all of the identity data values have been validated by a component, a secure communication

session may be established, and the components may be informed that the session may be fully trusted. When less than all of the identity data values have been authenticated, the components may continue to establish a secure communication session; however, the components may be informed that the session should only be partially trusted. As a result the components may be programmed to only permit a subset of operations to be performed responsive to the communication between the components.

In addition, exemplary embodiments of the previously described method may further include a unique session identification value or SessionID. A different SessionID may be generated for each attempt to establish a secure communications session between two components. Such a SessionID may be used to prevent a rouge application from recording handshaking communications between original components and replaying the communications to establish a secure communication with one of the original component.

Also, in further exemplary embodiments after the secret key has been successfully transferred between components, each of the components may be operative to independently produce identical but secret hashes of the secret key. The hashes may be used to encrypt and decrypt messages sent between the components. For each subsequent message a different hash derived from the original secret key may be generated for use in encrypting and decrypting each new message. As a result, even though the content of two or more messages may be identical, the corresponding encrypted form of those identical messages will be different.

Figure 5, shows an example of a system which uses more than one identity data such as a serial number to authenticate components and establish a secure communication session between the components. In this example one component corresponds to a software component 506

operating in a computer 502 of an ATM 500 and the other component corresponds to a hardware device such as a cash dispenser 504 of the ATM. In this example, the computer 502 of the ATM 500 may be located outside a secure chest or safe 512 of the ATM. The ATM may further include all or portions of the cash dispenser 504 located within the safe 512.

5           The computer may include at least one processor 514 and at least one hard drive 510. A software component 506 such as a device driver, service provider, or other device interfacing program may be stored on the hard drive and execute in the processor of the computer. The processor may be associated with a unique processor identification number or serial number (SN1). Also the hard drive may be associated with a unique hard drive identification number or  
10          serial number (SN2). Each of these serial numbers may be retrievable by the software component 506.

          The cash dispenser may include: a processor 516; a data store 518 such as a flash RAM or other nonvolatile memory; and a software or firmware cash dispenser component 522 that operates in the processor 516 of the cash dispenser.

15           In an exemplary embodiment of the ATM, the cash dispenser may be connected to the computer through a communication line 508. The communication line may correspond to a USB cable or other serial, network, or other communication line which is operative to transport data between the cash dispenser and the computer.

          The cash dispenser may include a public / private key pair stored in the data store of the  
20          cash dispenser. In an exemplary embodiment the cash dispenser component 522 may be operative to generate the public / private key pair and store the keys in the data store 518. In

alternative exemplary embodiments, the cash dispenser may be manufactured to include a unique public / private key pair in the data store.

In an exemplary embodiment, the cash dispenser may include an input device 520 such as a button which is located within the safe 512. Such an input device may only be accessed by a user by unlocking and opening a door to the safe. The cash dispenser may be responsive to the input device being activated by a user to accept one or more serial numbers sent to the cash dispenser from software operating in the computer 502 of the ATM. In this described exemplary embodiment, the serial numbers sent by the computer and accepted by the cash dispenser in response to a button 520 being pressed, may include the unique serial numbers (SN1, SN2) of the processor and hard drive of the computer. However, it is to be understood that in alternative exemplary embodiment, other serial numbers may be used. The cash dispenser may be operative to store the received serial numbers in the data store 518 of the cash dispenser.

In the described exemplary embodiment, the cash dispenser may be operative to determine if the button 520 is pressed. If the button is not pressed at the time the serial numbers are sent to the cash dispenser, the cash dispenser may reject or otherwise discard the received serial numbers. In alternative exemplary embodiments the cash dispenser in response to the button being pressed may send a message to the computer which causes the computer to send the serial numbers to the cash dispenser for storage in the cash dispenser.

Figure 6, shows a schematic view of the system. Here both the software component 506 and the cash dispenser component 522 may be operative to access a message protect application 524. In this described exemplary embodiment, each of the processors in the computer and the cash dispenser may execute individual instances of the message protect application 524.



The message protect application 524 may include one or more functions which are callable by the software component and cash dispenser component to perform the cryptographic calculations that are involved in an exemplary method for establishing a secure communication session between the components.

5           Prior to forming a secure communication session between the software component 506 and the cash dispenser component 522, each is operative to access the serial numbers (SN1, SN2) associated with the computer. In this described exemplary embodiment, the software component 506 of the computer may be operative to retrieve the serial numbers from the local hardware devices (i.e. the processor and hard drive). The cash dispenser component 522 may be operative  
10       to retrieve the serial numbers from the data store 518.

When establishing a connection, the software component 506 and the cash dispenser component may send their corresponding message protect applications the required data needed to generate the messages used to establish a secure communication session between the components.

15           For example in the described exemplary embodiment shown in Figure 6, the cash dispenser component is operative to generate and send a first message 600 to the software component. The first message may include the public key associated with the cash dispenser and a new SessionID generated by the cash dispenser component. A different SessionID may be randomly generated for each initial attempt to establish a secure communication session with  
20       another component.

In response to receiving the first message, the software component 506 is operative through calls to the message protect application 524 to generate a second message. As discussed

previously the software component may be operative to have a secret key randomly generated. Also as discussed previously the software component may be operative to have the secret key encrypted with the public key received in the first message.

5 In the previously described method the software component was further operative to have the identity data associated with the software component hashed with the public key. In this described exemplary embodiment, there may be more than one identity data. For example, the software component may be operative to retrieve two identity data values or serial numbers (SN1, SN2) associated with the computer. The software component may pass these two serial numbers (SN1, SN2) to the message protect application. The message protect application may  
10 be operative to generate a plurality of hashes using data that may include the two serial numbers (SN1, SN2), the public key (PublicKey) of the cash dispenser, and a further default serial number (DefaultSN). Different combination of this provided data may be hashed together using a one way hash algorithm such as MD5 for example to produce three authentication hashes. In an exemplary embodiment the following Equations show example formulas for generating the  
15 authentication hashes:

$$\text{Hash1} = \text{MD5} (\text{SN1} + \text{SN2} + \text{PublicKey}) \quad (1)$$

$$\text{Hash2} = \text{MD5} (\text{SN1} + \text{DefaultSN} + \text{PublicKey}) \quad (2)$$

$$\text{Hash3} = \text{MD5} (\text{DefaultSN} + \text{SN2} + \text{Public Key}) \quad (3)$$

20 In the exemplary embodiment, the second message generated by the software includes the encrypted secret key, the three authentication hashes and a copy of the SessionID received in

first message. To enable the information in the second message to be more secure, the three authentication hashes and/or the SessionID may further be encrypted using the secret key. In alternative exemplary embodiments a combination of the secret key, authentication hashes and/or the SessionID may be encrypted using the public key of the cash dispenser.

5           After the second message has been generated it is sent to the cash dispenser. The cash dispenser component may through calls to its message protect application 524 decrypt the information in the second message and authenticate the second message.

          For example, the cash dispenser component may retrieve the stored serial numbers (SN1 and SN2) and the public key and the private key of the cash dispenser from the data store of the cash dispenser. The serial numbers, public key, private key, and the second message may be  
10           passed to the message protect application. The original SessionID may also be passed to the message protect application. The message protect application may be operative to use the private key to decrypt the secret key in the second message. The message protect application may further use the secret key to decrypt the authentication hashes and/or the SessionID in the second  
15           message.

          To verify that the second message was generated in response to the first message, the message protect application may be operative to verify that the original SessionID sent in the first message corresponds to the SessionID received in the second message. If the SessionIDs do not match the message protect application may be operative to indicate to the cash dispenser  
20           component that the handshaking between the components has failed to produce a secure commendation session.

To verify that the second message originated from a trusted software application in the computer of the ATM, the message protect application may be operative to generate three further authentication hashes using the previously described Equations 1-3. These three further authentication hashes may be generated using the serial numbers (SN1, SN2) stored in the data store of the cash dispenser, the defaultSN, and the public key of the cash dispenser.

In an exemplary embodiment, the defaultSN may correspond to a value that is known or available for use by both instances of the message protect applications executed by the computer and the cash dispenser. Such a defaultSN may equal "1234567890", "ABCDEFGHJIJ" or some other known value. In this described exemplary embodiment, when the serial numbers used to form the authentication hashes in the second message correspond to the serial numbers known to the cash dispenser, all of the authentication hashes in the second message will match the further authentication hashes generated by the cash dispenser. In this case, the message protect application may be operative to indicate to the cash dispenser component that a secure communication session has been formed and that the session may be fully trusted.

If all of the authentication hashes in the second message do not match the further authentication hashes generated by the cash dispenser, then the message protect application may be operative to indicate to the cash dispenser component that a secure communication session has not been established. If one but not all of the authentication hashes in the second message match the further authentication hashes generated by the cash dispenser, then the message protect application may be operative to indicate to the cash dispenser component that a secure communication session has been established but may only be partially trusted. In alternative exemplary embodiments, based on which hashes fail to be identical, the message protect

application may indicate to the cash dispenser component which of the serial numbers did and/or did not match.

The cash dispenser component may perform all, none or a limited subset of operations requested by the software component in response to an indication that the secure communication session is only partially trusted. For example, if one but not both of the processor and hard drive on the computer or other components associated with the serial numbers (SN1, SN2) are replaced by a technician servicing the ATM, one of the two serial numbers used by the software component to generate the second message will be different than the corresponding serial number stored by the cash dispenser. In this case, a secure communication session may be formed which is only partially trusted. When at least one of the serial numbers remains the same, the cash dispenser may be adapted to continue to operate to dispense cash in response to commands received from the software component. Also, when only one of the serial numbers remains the same, the cash dispenser may be adapted to have reduced functionality which limits the functionality of the cash dispenser for example to only performing servicing and diagnostic functions in response to commands received from the software component. In further exemplary embodiments, the cash dispenser may be adapted to not establish and use the secure communication channel when at least one of the serial numbers is not valid.

In the exemplary embodiment, the cash dispenser component is operative to generate a third message 604 which is sent back to the software component. The third message may include status data representative of the trust level determined by the cash dispenser component so that the software component will be operative to determine what commands the cash dispenser will accept. For example, if the status data indicates that the secure communication session is fully

trusted, then the software component may be operative to send encrypted messages through the secure communication session which request the cash dispenser to dispense cash for example. If the software component corresponds to a device driver for example, the software component may then return a message to a calling program which indicates that the cash dispenser is ready to receive commands to dispense cash or perform other operations.

If the status data indicates that no secure communication session has been established, then the software component can propagate an error message to the calling program and/or an error log which indicates that the cash dispenser is not ready to receive commands. The error message may be generated in response to the status data and may indicate the reason for the error. Such reasons may include a failure to verify the serial numbers of the processor or hard drive for example. Such reasons may also indicate that an invalid SessionID has been detected

As described previously, once a secure communication session has been established, the two components (i.e. the software component and the cash dispenser component) may proceed to use the transferred secret key to encrypt and decrypt messages sent between them. In an exemplary embodiment, to further increase the security of the communications, each time a message is to be encrypted and passed between the two components, the two components may individually calculate new secret key hashes based on the original secret key transferred between the components. Each new secret key hash generated may only be used to encrypt and decrypt a single message. For the next message and each subsequent message, new modified secret key hashes (i.e. nonce) may be generated by the software component and the cash dispenser. Because the new secret key hashes are derived from the original transferred secret key, both the software component and the cash dispenser component are operative to generate the same sequence of

new secret key hashes for each new message and as a result may continue to encrypt and decrypt the new messages sent between them.

For example, in an exemplary embodiment once the secure communication session has been established, future communications may use MD5 and the Advanced Encryption Standard (AES) cryptographic algorithm along with different secret key hashes to ensure communications are not duplicatable. In an exemplary embodiment, the secure communications may be comprised of two parts: a message header and encrypted message data. The message header may include a message authentication code (MAC) for the message data. Such a MAC may be generated by creating a first message digest of the message data. In an exemplary embodiment a first message digest may be formed by hashing the message data with a secret key hash using a one-way hashing function such as MD5. Further, a second message digest may be generated by hashing the first message digest with the secret key hash. In this described exemplary embodiment the resulting second message digest may correspond to the message header.

The encrypted message data may be formed by passing the message data and the secret key hash through a symmetrical encryption algorithm such as AES. The messages communicated between components such as the computer and the cash dispenser may then include the combination of the generated message header and the encrypted message data.

The component receiving the message may be operative to decrypt the message data using a secret key hash derived from the original transferred secret key. To authenticate the message, the receiving component may also generate a MAC from the message data which may then be compared to the message header.

As discussed previously, the secret key hash may be modified with each newly sent and received encrypted message. Thus for the next message sent between the two components, a new secret key hash may be used which is generated by incrementing the previously secret key hash with a known value. In exemplary embodiments, different sets of new secret key hashes may be generated for both sending and receiving by the components.

In the previously described exemplary embodiments, the authentication hashes are generated by hashing one or more identity data values such as serial numbers with a public key of a component. The same public key may also be used to encrypt the randomly generated secret key. However, in alternative exemplary embodiments, the identity data values such as serial numbers may be hashed with another public key or other shared hashing argument (HashingArg) which is different than the public key used to encrypt the randomly generated secret key. In such an alternative exemplary embodiment, the component initiating the secure communication session may generate a first message 600 which includes the public key for use in encrypting the secret key, the unique SessionID, and a further hashing argument which is used in place of the public key in the previously described Equations (1-3). The corresponding equations used for generating the authentication hashes may be as follows:

$$\text{Hash1} = \text{MD5} (\text{SN1} + \text{SN2} + \text{HashingArg}) \quad (4)$$

$$\text{Hash2} = \text{MD5} (\text{SN1} + \text{DefaultSN} + \text{HashingArg}) \quad (5)$$

$$\text{Hash3} = \text{MD5} (\text{DefaultSN} + \text{SN2} + \text{HashingArg}) \quad (6)$$



In further alternative exemplary embodiment, the unique sessionID may be used for the hashing argument in place of the public key. As a result the equations for generating the authentication hashes may be as follows:

$$\text{Hash1} = \text{MD5} (\text{SN1} + \text{SN2} + \text{SessionID}) \quad (7)$$

$$5 \quad \text{Hash2} = \text{MD5} (\text{SN1} + \text{DefaultSN} + \text{SessionID}) \quad (8)$$

$$\text{Hash3} = \text{MD5} (\text{DefaultSN} + \text{SN2} + \text{SessionID}) \quad (9)$$

10 In this described exemplary embodiment, the SessionID may not be included in the message which sends the encrypted secret key and the authentication hashes. Rather the further authentication hashes may be generated using the original sessionID. As a result, the received authentication hashes and the further authentication hashes generated will only match when the same sessionID was used to generate both sets of authentication hashes.

15 The described exemplary embodiment shows a cash dispenser of an ATM that is operative to establish a secure communication channel with a software component of the computer of an ATM. However, it is to be understood, that in alternative exemplary embodiments, that other software and hardware components in the ATM be use the same methods to establish a secure communication session between them. For example in alterative exemplary embodiments, software components such as device drivers may be operative to perform the above described methods to established a secure communication session with a cash recycler, depository, or other devices in the ATM. Further the described methods may be used to

establish a secure communication session between a host banking system or other remote system and an ATM.

It is to be understood that the previously described exemplary embodiments of methods for establishing secure communication session between components may or may not use dedicated cryptography hardware or chips such as a TPM to perform cryptographic calculations. Cryptographic calculations for example may be performed using software programs such as the message protect application running in the processors of the computer and hardware devices.

However, as discussed previously, alternative exemplary embodiments of an automated banking machine may use cryptographic hardware such as a TPM to establish secure communications between components using all or portions of the previously described methods and/or other methods for establishing secure communications and trust between components.

For example, an alternative exemplary embodiment of an ATM may include a trusted platform (TP) which may also be referred to as a trusted computing platform or a trusted computer base (TCB). Figure 7 shows an exemplary embodiment of a trusted platform 300 for an ATM 310. The ATM 310 may include a computer 312. The computer 312 may include a motherboard with a BIOS 316. The computer 312 may further include a trusted platform module (TPM) 314 that complies with a trusted platform specification such as the Trusted Computing Platform Alliance (TCPA) specification, Microsoft next-generation secure computing base (NGSCB) (formerly called Palladium) or other specification for establishing a trusted and secure computing environment.

In exemplary embodiments, the TPM may be comprised of one or more computer chips integrated into the motherboard of the computer 312. An example of a TPM which may be used

with exemplary embodiments of an ATM includes a TPM SLD 9630 chip of Infineon Technologies AG which is based on the TPCA Main Specification version 1.1b. Also, in exemplary embodiments, the TPM may correspond to Microsoft's specification for a security support component (SSC) for the Microsoft NGSCB. In alternative exemplary embodiments the TPM may also be incorporated into the processor of the computer of the ATM, or a dongle, hardware card, or other device in operative connection with the computer of the ATM

As used herein a trusted platform (TP) corresponds to hardware and software components of a computer that conform to a trusted platform specification such as Microsoft's NGSCB, the TPCA specification, or another trusted platform which uses a combination of a TPM and one or more trusted software components.

As used herein the TP may be described as being used to perform certain functions which may require the use of a TPM. However, it is to be understood that in addition to the TPM, software associated with the TPM may be involved in the performance of the functions. Such software may include a trusted portion or kernel of the operating system such as a nexus in Microsoft's NGSCB specification. Such software may further include a trusted application, nexus computing agent (NCA), application programming interface (API), or other software component that enables the TP to perform the described function. As used herein and in the claims references to using a TP to perform a particular cryptographic, authentication, or other function may include the use of the TPM chip and/or software which requires a TPM chip such as a nexus or NCA.

The ATM 310 shown in Figure 7 includes a TP 300 that conforms to the TPCA Main Specification version 1.1b. Here the ATM 310 may include a TPCA Software Stack (TSS) 318

which may provide modules and components that provide the supporting functionality to the TPM. The modules and components of the TSS may support the offloading of security functions from the TPM to the main CPU(s) of the computer of the ATM

The ATM 310 may further include a secure API 320 which provides a set of functions that ATM software 322 may use to configure and communicate with the TPM through the TSS.

In an exemplary embodiment the secure API may include functions for accessing on-chip cryptographic hardware functions 330 of the TPM such as encryption, decryption hashing, and key generation. Examples of on-chip encryption and decryption functions that may be accessed through the secure API may include RSA, DES, and Triple DES for example. In the exemplary embodiment the terminal control software 324 may perform encryption and decryption operations involved with the performance of transactions for consumers using the on-chip functions of the TPM. In alternative exemplary embodiments the ATM software may be operative to communicate with the TPM through direct access to the functions of the TSS. In further alternative exemplary embodiments, the ATM software may be operative to communicate with the TPM through a Microsoft cryptography API (CAPI) included with a Microsoft operating system installed in the ATM for example.

Figure 8 shows an exemplary embodiment of a TP 400 for an ATM 410 which conforms to the Microsoft's NGSCB specification. Here the TPM chip 414 may correspond to or may be referred to as a security support component (SSC). Also, with this specification the software which interfaces with the TPM chip may include the NGSCB nexus 420 or other secure kernel of an operating system 418. An ATM with a NGSCB trusted computing platform may include two modes of operation referred to as the standard mode 430 and the nexus mode 432. Both modes

may be operative at the same time. Legacy applications which are not designed for use with a TPM 414 may execute in the standard mode of the platform. However, trusted applications which are designed to use the TPM may be granted permission to operate in the nexus mode 432. Such trusted applications include the NGSCB nexus 420 and one or nexus computing agents (NCBs).

In an exemplary embodiment of the ATM 410, at least one of the terminal control software components 422 of the ATM may be granted permission to operate in the nexus mode. In a TP based on Microsoft's NGSCB specification such components may be programmed to use features of the TP through communication with the nexus 420. As used herein, terminal control software components which interface with a TPM and/or which operate in the nexus mode are referred to as trusted ATM components 440. In a TP based on Microsoft's NGSCB such trusted ATM components correspond to NCBs.

Although, in an exemplary embodiment of the ATM, all of the terminal control software may be programmed to operate in the nexus mode, in other exemplary embodiments, one or more of the terminal control software components may continue to operate in the standard mode 430 of the machine. For example, legacy ATM software components 442 such as a WOSA/XFS (Windows Open Services Architecture/eXtensions for Financial Services) manager or other XFS layer or other device interface layer may continue to operate in the standard mode. However, other components, such as software components which have access to secure financial information, items of value ( i.e. cash, deposits) for example may operate on the nexus mode of the TP. Examples of such trusted ATM components may include components which communicate financial information such as account data to a remote host banking system. Other

examples of trusted ATM components may include service providers, device drivers, or other portions of the device interface layer which communicate with input devices, output devices, and transaction function devices such as a cash dispenser, cash recycler, or depository mechanism. In addition trusted ATM components may include the terminal applications and/or configuration components for use with servicing, installing, maintaining or configuring the ATM.

In this described exemplary embodiment, trusted ATM components may use the following mechanisms of a TP: curtailed execution, sealed storage, secure I/O, and attestation. With curtailed execution, a trusted ATM component is isolated from software that is not trusted by the trusted ATM component. With sealed storage, the trusted ATM component has access to secret information that is not available to other software. With attestation, the trusted ATM component has access to an authentication mechanism which enables the component to prove its identity to other software components located on the ATM or remote from the ATM. With secure I/O, the trusted ATM component is operative to receive secure inputs or send secure outputs through secure input and output devices of the ATM.

In the exemplary embodiment of the ATM, such secure input and output devices may include, a secure keyboard and video controller. In addition secure input and output devices may include keypads, card readers, receipt printer and/or any other device that is adapted to securely receive or transmit inputs and outputs between a user and the ATM.

The TP may be used to provide secure storage for symmetric and asymmetric keys and other protected information used by the trusted ATM components. The TP may also be used to perform cryptographic processes such asymmetric and symmetric encryption and decryption for trusted ATM components. The TP may also be used to perform key generation, public/private

key pair generation, random number generation, hash generation, digital signature generation, and digital signature verification for trusted ATM components. In addition The TP may be used to provide a report or attestation to the current configuration of the ATM or other information. For example, the TP may be used in acquiring and/or verifying information about trusted ATM components and hardware device installed on the ATM.

As used herein, the information acquired about a component of the ATM for attestation using the TP is referred to as a measurement of the component. Such a measurement may take place when the ATM is initially booted or at other times during the operation of the ATM. The measurements may be securely stored using the TP and used later to determine if the configuration of the ATM has changed.

In an exemplary embodiment, a trusted ATM component and/or an external system remote from the ATM such as a host banking system may be operative to verify the information about the measured components of the ATM. The exemplary embodiment of the TP may be used to attest to the information about the measured component by digitally signing the measurement of the component with a private key of the TPM. The digitally signed measurement may be sent to the software application, device, and/or external system requesting information about the measured component.

In the exemplary embodiment the TPM may include at least one private key securely stored in the TPM chip. The TPM may further include at least one digital certificate that includes the public key which corresponds to the private key stored in the TPM. Such a digital certificate may be signed by the motherboard or TPM manufacturer for example.

The TPM may also be operative to store additional public and private keys and corresponding digital certificates. The additional digital certificates associated with the TPM may be signed by a certificate authority which enables them to be verified by authenticating the signature of the certificate authority. In exemplary embodiments, the TPM may be operative to provide a root of trust for the ATM through a PKI infrastructure. The TPM may be operative to extend its trust to one or more components of the ATM, such as a nexus and trusted ATM components, by building a chain of trust. Also trusted ATM components may be operative to extend their trust to other components.

In exemplary embodiments, the TP of the ATM may be used to enable the computer of the ATM to boot into a known state with hardware and software that is verified through use of the TPM. In this described exemplary embodiment, the ATM software 322 such as the terminal control software may include trusted ATM components that are operative to audit the configuration of the ATM. Such auditing ATM components may through use of the TP acquire measurements for a plurality of the software and/or hardware components of the ATM. The results of the measurements may be compared to known authorized configurations of valid measurements of the components. In an exemplary embodiment such authorized configurations may be digitally signed to prevent undetected unauthorized modifications to the authorized configurations.

In alternative exemplary embodiments, the authorized configurations may be stored in a data store managed by a remote system such as a server associated with a host banking system. Prior to authorizing the ATM to perform transaction functions, the host banking system may communicate with the ATM to direct that the trusted ATM components acquire measurements of



one or more devices of the ATM. The measurements may be returned to the host banking system for verification against the known authorized configurations for the ATM. If the measurements do not correspond to valid measurements of one or more authorized configurations for the ATM, the host banking system may be operative to prevent the ATM from performing transactions.

5           In the exemplary embodiment, the TP may be used to measure components as soon as the ATM is booted. If the hardware and/or software components are not successfully verified, the TP may be operatively configured to prevent the computer of the ATM from booting into a fully functional state. For example, the TPM may be operative to verify that the BIOS of the computer motherboard has not been altered. The TPM may be operative to measure the nexus or  
10   other kernel of an operating system before enabling the nexus or other kernel of an operating system to execute. For example, the measurement may include loading a nexus into secured curtailed memory and sending an image of the memory to the TPM chip. The TPM chip may make a cryptographic hash of the image. When the nexus is first started, this hash may be permanently stored in the TPM chip. On subsequent boots of the ATM, the TPM chip may be  
15   operative to measure the current nexus and verify that the hash of the nexus matches the hash originally stored in the TPM chip.

          In exemplary embodiments, the TP may also be used to measure hardware configurations and hardware devices to verify that the ATM includes the correct hardware. The TP may also be used to measure the installed software of the ATM such as the operating system and other  
20   specified software to assure that the installation of the ATM has not had an unauthorized change. The software being verified may include both trusted ATM components and other software components and applications which may not run on the nexus mode of the TP.

In the exemplary embodiment the TPM may be used to manage secure and protected data stores in the ATM. Such secure and protected data stores may include an internal data store 340 such as a nonvolatile RAM located on the TPM. Such secure and protected data stores may also include a data store 342 which is external to the TPM. Information stored in the data stores may be encrypted by the TP using public keys and an RSA cryptography algorithm and/or symmetric keys using a symmetric cryptography algorithm.

In exemplary embodiments, information used by an ATM such as secret keys, signature keys, verification keys, encipherment keys, decipherment keys, private keys, public keys, and authorization keys or other critical information used in the operation of the ATM may be securely stored in the ATM through use of the TP. For example, secret keys used by an encrypting pin pad (EPP) may be stored in an internal or external data store managed through use of the TP. Also private keys used for creating digital signatures for the ATM may be securely stored using the TP. The ATM may use the TP to sign messages using the private keys stored by the TPM. Examples of other information that may be securely stored in the ATM using the TP may include information regulated by government or industry such as anti-fraud mechanism information for areas such as bulk note tables, canister mappings, account access, transaction journals, and repudiation.

In an exemplary embodiment, information received by the TP from a trusted ATM component may be stored in an encrypted form in a data store using a key associated with the trusted ATM component. Such a trusted ATM component may correspond to the owner of the information. The exemplary embodiment of the TP may enable the owner trusted ATM component to use the TP to retrieve and return the information stored in the data store back to the

owner component. The exemplary embodiment of the TP may be operative to verify that the request is from the owner component. If the request comes from a component that does not correspond to the owner of the information, the TP may be operative to reject the request for the information.

5           Automated banking machines such as an ATM may be operative to perform cryptographic calculations involved with securely sending and receiving an ATM terminal master key, ATM communication keys, and user entered personal identification numbers (PINs) between the ATM and an ATM host banking system. Examples of ATM systems which use cryptographic calculations and protocols to securely transfer keys between an encrypted  
10   pin pad of an ATM and a host banking system and which may be adapted to use a TP are shown in U.S. application serial number 10/126,140 filed April 19, 2002 which is incorporated herein by reference in its entirety. In an exemplary embodiment of the ATM, one or more cryptographic calculations required by the EPP described in application serial number 10/126,140 may be performed using the on-chip or off-chip cryptographic functions of the TP.

15           For example, in an exemplary embodiment, a service technician may initiate a function at the ATM or from a remote location which causes a trusted ATM component to send a request to a host banking system for a terminal master key. The request may include a digital certificate associated with the TP and/or the trusted ATM component. The host banking system may validate the certificate and use the public key associated with the certificate to encrypt a terminal  
20   master key. The host banking system may then send a message to the ATM which includes the encrypted terminal master key. Once the message is received, the trusted ATM component may

be used to decrypt the encrypted terminal master key using a corresponding private key and an asymmetric cryptography algorithm such as RSA.

The message from the host banking system with the encrypted terminal master key may be digitally signed by the host banking system. The trusted ATM component may validate the terminal master key through verification of the digital signature. In an exemplary embodiment, the unencrypted terminal master key may be securely transferred to an encrypted pin pad for use with decrypting symmetrical communication keys provided to the EPP by the host banking system.

As a further verification, the ATM may output a hash of the public key of the host system which was used to validate the signature of the host banking system. The ATM may require a technician to provide an input to the ATM which is representative of a confirmation that the hash of the public key of the host banking system is valid. In addition the public key of the host banking system may be included on a digital certificate of the host banking system. To enable the trusted ATM component to further verify the public key of the host banking system, the host digital certificate may be signed by a certificate authority trusted by the TP and/or the trusted ATM component. The trusted ATM component may verify the digital signature of the certificate authority to verify the public key included in the digital certificate of the host banking system.

In an alternative exemplary embodiment, the terminal master key may remain securely stored by the TP in a protected data store managed by the TP rather than sending the terminal master key to an EPP. The TP and/or a trusted ATM component may be used to decrypt communication keys provided to the ATM by the host banking system using the stored terminal master key. The TP and/or trusted ATM component may then securely transfer the

communication key to the EPP. The EPP may use the communication key to encrypt personal identification numbers (PINs) inputted into a key pad input device of the EPP by a user performing a transaction at the ATM. The encrypted PIN may then be sent to the automated banking machine for use in authorizing the transaction to be performed.

5           In a further alternative exemplary embodiment, the communication key may remain securely stored by the TP in a protected data store managed by the TP rather than sending the communication key to an EPP. The TP and/or trusted ATM component may then use the stored communication key to encrypt PINs inputted into an input device other than an EPP of the ATM prior to sending the PINs to the host banking system for authentication. Such input devices may  
10           correspond to a keypad, keyboard, touch screen of a display device, or any other input device of the ATM. In this described alternative exemplary embodiment, the input devices may be operatively configured to securely communicate an inputted PIN to the TPM and/or trusted ATM component by establishing a secure encrypted communication channel between the input device and TP and/or trusted ATM component.

15           In exemplary embodiments where the cryptographic functions on the TPM chip are not sufficient for the cryptography requirements of the ATM, the exemplary TP may be operative to securely manage cryptographic calculations using protected cryptographic software and/or hardware components 332 which are external to the TPM. Such external cryptographic  
components 322 may include cryptography calculations performed by the CPU(s) of the  
20           computer of the ATM responsive to the TPM, nexus, TSS, a cryptography API of the Operating system, and/or a trusted ATM component.

In this described exemplary embodiment, the TP may be used to measure and verify the EPP device in the ATM and software which operates the EPP prior to enabling the ATM to perform transaction functions using the EPP. In addition the TP may be used to measure and verify other hardware devices and software devices of the ATM prior to enabling the ATM to perform transaction functions using the hardware or software components. For example, the TP may be used to measure and verify that valid transaction function devices are installed in the ATM such as a cash dispenser and depository mechanism. The TP may also be used to measure and verify that other types of hardware devices of the ATM are valid such a card reader device, display screen, function keys, keypads, network card, hard drives or any other device that may be included in an ATM. In addition the TP may be used to measure and verify software components used to control the hardware devices of the ATM. Examples of such software devices which may be measured and verified by the TPM may include device drivers, configuration files, service providers, databases, XFS layers, ODS layers, TEC components, browser software, JVMs, terminal control software components, diagnostic software applications, server applications, service and maintenance software applications or any other software components including software files and data that may be included in an ATM.

In the exemplary embodiment, the measurement, attestation, cryptographic functions and protected storage features of the TP may be used to establish secure communications between hardware and software components of the ATM in which components can verify each other and authenticate the communications passed between them. In addition, in exemplary embodiments, where the ATM includes legacy components which do not include a mechanism for establishing secure communications, the TP may be used to minimize the security risks such legacy

components may pose to the ATM. For example ATMs may include an XFS layer which by itself may have limited functionality for securely controlling devices of the ATM. A TP may be used to measure and verify both the XFS layer and components that receive and send communications through an XFS layer of an ATM. Examples of an ATM system which is operative to establish secure communications between components of an XFS enabled ATM and which may be adapted to use a TP to establish such secure communications are shown in U.S. application serial number 60/429,250 filed November 25, 2002 which is incorporated herein by reference in its entirety.

For example, XFS enabled ATMs may include application layer components. Such application layer components may be operative to control XFS compatible hardware devices through communication with an XFS layer. In this described exemplary embodiment the application layer components above the XFS layer may correspond to trusted ATM components. In addition one or more of the device driver layer components or other software interfaces between the XFS layer and the hardware components may also be trusted ATM components.

Examples of device driver layer components include device drivers, service providers, and module interfaces such as a unified base release (UBR) components or an Agilis module interface (AMI).

In an exemplary embodiment, to make communications with the XFS layer more secure, the device driver layer components may only be responsive to control hardware devices when the communication received from the XFS layer has been confirmed to originate from a trusted ATM component. In this exemplary embodiment, a security component of the ATM may be operative

to supply this confirmation to the device driver layer components. For example, each time an application layer component communicates through the XFS layer, the application layer component may also notify the security manager component. The security manager component may then confirm to a device driver layer component that the communication received through the XFS layer originated from an authorized trusted ATM component. In this described exemplary embodiment, the application layer components, device driver layer components, and the security manager may use the attestation features of the TP to prove their identities to each other and to establish secure communication between them.

For example, a first application layer component may send a first communication to a security manager component of the ATM. The first communication may be attested to through use of the TP to enable the security manager component to authenticate the first communication. The application layer component may then send a second communication through the XFS layer of the ATM which is directed to the device driver layer of the ATM. In this described exemplary embodiment, the device driver layer may cause a hardware device of the ATM to operate responsive to both the second communication and the security manager.

In an exemplary embodiment the device driver layer may communicate with the security manager to verify that it may operate the hardware device responsive to the second communication. The communications from the security manager may be attested to through use of the TP to enable the device driver layer component to authenticate communications from the security manager.

In addition to using the TP to authenticate communications passed through an XFS layer, the TP may also be used to measure and verify that the software stack that corresponds to the



components of the XFS layer, application layer and device driver layer are valid and have not been changed to an unauthorized configuration.

In an exemplary embodiment, the service software of the ATM may be operative to install and manage digital certificates used in the operation of the TP. Examples of digital certificates that may be managed on the ATM by the service software includes: validation certificate, TPM endorsement certificate, TPM identity certificate, conformance certificate, and platform certificate. In the exemplary embodiment, the validation certificate may include data that includes the TPM validation data statements and labeling. The endorsement certificate may be used to vouch for the existence of a particular TPM. The TPM identity certificate may include a public key for a TPM identity and a description of the TPM and platform. The conformance certificate may be issued by an entity chosen by the manufacture of the platform to vouch for the platform meeting the requirements of the specification for the ATM. A platform certificate may attest that the ATM platform has been built according to manufacture of the ATM.

In the exemplary embodiment the service software of the ATM may include ownership management functions for managing a transition in ownership of the ATM: from the manufacture of the ATM to the entity responsible for installation of the ATM; to the institution or entity that operates the ATM; and to the entity that is responsible for maintenance of the ATM. The service software may be operative to manage in a trusted manner the identities and status settings for the TP as the ATM transitions from one entity to another.

The service software of the ATM may also be operative to perform secure backup and recovery functions of information stored on the TPM chip and/or other protected storage locations managed by the TP. Such backups of information managed by the TP may also be

securely stored on a system remote from the ATM in an encrypted form which can not be deciphered by the remote system. The service software of the ATM may also be operative to securely migrate selected information managed by a TP of an ATM to a TP of a trusted platform to support maintenance of the ATM.

5           In the exemplary embodiment the service software may include functions for managing changes to the TP and the trusted ATM applications. The service software 326 may modify the authorized states of the ATM that are measured and/or verified by the TP. The service software may enable different portions of the ATM to be configured based on the type of user performing the task. For example, an administrative user may be permitted to update the TP to trust new  
10 hardware and/or software installed on the ATM, while a maintenance user may be limited to updating a specific subset of the software and/or hardware configurations to be trusted by the TP. In the exemplary embodiment the TP may also be used to maintain an audit trail of all changed to the configuration of the ATM. Audit trail logs for operations of the TP and hardware and software components of the ATM may be securely stored in a local or external data store  
15 managed by the TP.

          The exemplary embodiment of the ATM may be operative to prevent unauthorized software and hardware from being installed on the ATM. If such software is installed on the ATM, the TP may be used by the ATM to determine that the system has been modified or changed to an unauthorized software and/or hardware configuration. In an exemplary  
20 embodiment, when the system is modified to an unauthorized configuration, the TP may be used to prevent the ATM from booting, and/or loading into a fully enabled state.

As discussed previously, the TPM chip may include a digital certificate signed by the manufacture or supplier of the TPM or computer motherboard. Also as discussed previously the TP may also be used to acquire additional digital certificates for the ATM. Such additional certificates may be issued or signed by a specific certificate authority that is trusted by one or more trusted ATM components of the ATM. An exemplary embodiment, the trusted ATM components may require the ATM to have been issued one or more digital certificates from the specific certificate authority prior to becoming fully functional on the ATM. In addition, systems remote from the ATM such as a host banking system, financial transaction server, or other server, may also require the ATM to be issued one or more digital certificates from a specific certificate authority prior to performing functions with the ATM.

For example, a trusted ATM component of the ATM may include a device driver such as a service provider (SP) which controls the operation of a cash dispenser. Such a cash dispenser SP may only be operative to cause the cash dispenser to dispense cash, after the cash dispenser SP has verified that the TP has possession of a particular private key. Such a private key may correspond to a public key named in a digital certificate issued by a specific certificate authority that is trusted by the cash dispenser SP. In an exemplary embodiment, communications received by the cash dispenser SP may be digitally signed or attested to by the TP using the private key. If the public key associated with the digital certificate issued by the specific certificate authority is usable to verify the digitally signed or attested to communication, then the cash dispenser SP may become operative to operate the cash dispenser in the ATM.

Although, in this example, an SP has been described as being operative to authenticate the TP, it is to be understood that in exemplary embodiments other trusted ATM components may be operative to make this verification before enabling SPs to operate hardware devices. For example in addition to service providers, the device driver layer of the ATM may include a module interface layer such as a UBR or an AMI which provides an additional layer of abstraction between the service providers and the hardware devices. One or more of the service providers may be programed to control hardware devices through communication with the module interface components rather than directly communicating with one or more hardware devices. In this described embodiment, the module interface may be adapted to authenticate the TP before enabling one or more SPs to operate their respective hardware devices.

In exemplary embodiments, the software associated with the TP may be operative to specify which SPs or other ATM components are trusted and enabled to operate in the ATM. The ATM may include a user interface which enables an authorized user to grant SPs or other trusted ATM components permission to operate using the TP. Each service provider or other trusted ATM component may include a manifest which includes a hash of the code of the program(s) or a public key(s) usable to verify digital signatures of the code of the program(s). The TP may be operative to authenticate each SP or other trusted ATM component using the information in the manifest. The manifest in turn may include a digital signature which can be authenticated by the TP to enable the SP or other trusted ATM component to operate in the ATM.

In addition to including software which specifies which applications installed on the ATM are trusted by and/or are granted permission to use the TP of the ATM, the ATM may

include configuration applications which are operative to update the ATM with new authorized software configurations of the ATM responsive to a license or other accessed or inputted set of rules. Examples of an ATM platform which is operative to authorize the installation and/or configuration of software on the ATM and which may be adapted to use a TP to authenticate an installation and/or configuration of an ATM are shown in U.S. application serial numbers 60/432,325 filed December 10, 2002 and 09/957,982 filed September 21, 2001 which are incorporated herein by reference in their entirety. In the exemplary embodiments, license information for software being installed on an ATM may be securely stored in an internal or external data store which is managed by the TP of the ATM.

Examples of license information which may be securely stored in an internal or external data store managed by the TP may include: configuration rules for software applications and hardware devices of the ATM, software application expiration dates, the name of the customer, or entity licensing the software, serial numbers for the software application, version numbers for the software applications, authorization keys for the software applications, softwareIds, hardwareIds, or any other information which is associated with the ATM. Each time an ATM is booted and/or an application of the ATM is started in the ATM, the license information securely managed in the TP may be compared to the state of the ATM as measured using the TP of the ATM. If the current measured state of the ATM does not correspond to the license information, the TP and/or ATM software may be operative to prevent the ATM from booting, and/or prevent the ATM software from running in an enabled state.

For example, in an exemplary embodiment, when new software is being installed in the ATM, the installation program may require that a technician input through an input device of the

ATM a first authorization key and license information. The installation program may use the TP to generate a second authorization key responsive to a secret key stored in the machine and the inputted license information. The inputted first authorization key may then be compared to the generated second authorization key. If the inputted first authorization key corresponds to the generated second authorization key, the installation software is operative to enable the new software application associated with the license information to be installed in ATM. In the exemplary embodiment the secret key used to generate the second authorization key may be securely embedded in the terminal control software, the TP or other secure location in the ATM. Also as will be explained below in more detail, the secret key may itself be generated responsive to keying material stored in one or more components of the ATM.

The process of installing the new software may include granting any new or updated trusted ATM components with permission to use and/or be trusted by the TP. For example, trusted ATM components may be granted permission to operate in the nexus mode. Also the TP may be updated to include authorized measurements for the newly installed software.

In exemplary embodiments, hardware devices of an ATM may include firmware which is undatable. As an unauthorized update to the firmware of a device may be performed, an exemplary embodiment of the ATM may be operative to authenticate the firmware content of a hardware device using a challenge and response mechanism. In an exemplary embodiment the TP of the ATM may be used to store a measurement of the firmware content. The measurement may then be compared to a known authorized value. If the firmware content does not correspond to the known authorized value, the exemplary embodiment of the ATM may be operative to prevent the ATM from using the device and/or operating in an enabled mode.

In exemplary embodiments, the ATM may be operative to measure the firmware of one or more hardware devices responsive to a request or challenge from a host banking system responsible for authorizing transaction functions performed at the ATM. The measurement may be reported back to the host banking system. The host banking system may compare the measurements of the firmware to known authorized values to determine if the firmware is valid prior to enabling or authorizing the ATM to perform transaction functions for users. As discussed previously, in exemplary embodiments, the measurement of the firmware may be measured using a TP of the ATM. However, it is to be understood that in alternative exemplary embodiments of the ATM, measurements may be taken by a further software or hardware device of the ATM and may be reported to the host banking system without using a TP.

An exemplary embodiment for measuring and reporting firmware content of a hardware device may include sending a measurement request message to the hardware device which prompts the hardware device to return a firmware certificate (FWC). The message may include an initial value which is used by the hardware device to generate the firmware certificate (FWC). A different initial value (INV) may be sent to the hardware device each time a measurement is requested. The hardware device may be operative to calculate the firmware certificate (FWC) responsive to the Initial Value (INV) so as to produce a different firmware certificate for each different Initial Value (INV) received with a measurement request message.

In the exemplary embodiment, the hardware device is operative to pass the entire content of the firmware, the Initial Value (INV), and a secret key (K) through a cryptographic algorithm which is operative to generate the firmware certificate (FWC). The firmware certificate cryptographic algorithm may be operative to generate a message authentication code (MAC) for

use as the firmware certificate (FWC). In exemplary embodiments the cryptographic algorithm may include one or more one way hash function such as MD5 or SHA-1. In a further exemplary embodiment, the firmware certificate cryptographic algorithm may correspond to Cipher Block Chaining (CBC) or a Cipher-Feedback Mode (CFB) MAC algorithm where the firmware content is encrypted with a block algorithm in CBC or CFB modes. The firmware certificate (FWC) may then correspond to the last encrypted block which may be encrypted once more in CBC or CFB modes. In other exemplary embodiments other cryptographic algorithms may be used to generate a unique firmware certificate (FWC) responsive to the firmware content, a secret key (K) and an initial value (INV).

For example, the firmware content may be comprised of a number of bytes. To calculate the firmware certificate (FWC) the firmware content may be divided into blocks of bytes, called firmware blocks (FWB), numbered from 0 to  $n-1$ . In an exemplary embodiment a block may be comprised of 8 bytes for example. As a result for a firmware content with 8k bytes, the firmware blocks (FWBs) may be numbers 0 to 1023. The firmware blocks (FWBs) may be encrypted in CBC mode using a symmetrical encryption algorithm such as DES. As shown in equation (10), an initial value included in the measurement request message to the hardware device may be XORed with the first block (FWB0) of the firmware content.

$$C0 = \text{DES}(K, \text{FWB0} \text{ xor } \text{INV}) \quad (10)$$

The result of the XOR may be encrypted with DES using a secret key (K) to produce a first cryptogram (C0). As shown in equation (11), each additional firmware block (FWBi) may be



ZORed with the previous calculated cryptogram (Ci-1). The result of the XOR may be encrypted with DES using the secret key (K) to produce another cryptogram (Ci).

$$C_i = \text{DES}(K, C_{i-1} \text{ xor FWBi}) \quad (11)$$

Equation 11 may be repeated until the last block but one. The firmware certificate (FWC) may be taken from a portion of the last cryptogram (Cn-2) such as the 6 rightmost digits of the last cryptogram (Cn-2). The firmware may then return the generated firmware certificate (FWC) to the component or system that issued the measurement request message.

In an exemplary embodiment of an ATM that includes a TP, the firmware certificate may be securely stored by the TP and compared to a known authorized value for the firmware certificate. Also, as described previously, the firmware certificate may be reported to a host banking system for comparisons by the host banking system to an authorized value for the firmware certificate. Such a known authorized value may be calculated in the same manner as described previously for the firmware using the same cryptographic algorithm that is responsive to a known authorized firmware content, the secret key, and the initial value to generate a firmware certificate. The received firmware certificate may then be compared to the generated firmware certificate to validate the firmware content in the hardware device.

In the exemplary embodiment of the ATM which includes the TP, the TP may be used to store the measurement of the firmware. Responsive to a call from another component or the host banking system, the TP may attest to the measurement by signing the measurement with a private key of the TP. The digitally signed measurement may then be forwarded to the calling

- component or host banking system. The calling component or host banking system may then validate the digital signature of the TP and authenticate the measurement by generating a firmware certificate to compare to the measured firmware certificate as previously described. In alternative exemplary embodiment, the ATM may use the TP to generate the firmware certificate  
5 for use in comparing with the measured firmware certificate.

In a further exemplary embodiment the firmware certificate may be generated by passing the firmware content through a one way hash algorithm such as MD5 and SHA-1. The resulting one way hash digest may then be combined with the initial value (INV) received in the measurement request message. The combined digest and initial value may then be encrypted  
10 with DES using the secret key (K) to produce a firmware certificate that varies responsive to each different initial value. In this described exemplary embodiment, to further increase the security of the firmware certificate, the combination of the digest and initial value may also be hashed with a one way hash function prior to encrypting the result with DES. The resulting firmware certificate may be validated using the TP, host banking system, or other component by taking a  
15 known valid one way hash digest of the firmware, combining it with the initial value (INV) and encrypting it using DES and the secret key. The measured firmware certificate and the value generated responsive to a know hash of the firmware may then be compared to validate the hardware device.

The previously described exemplary methods of measuring firmware content may also be  
20 used to measure other groupings of bytes including software files or other groupings of data stored on a hard drive or other data store of the ATM. However, it is to be understood that other measurements may be taken in different forms. For example measurements may correspond to

the reading of embedded hardware or software IDs in the components. Measurements may also correspond to having a component digitally sign a random number or other information to verify that the component has possession of a private key. Also as discussed previously measurements may also be made by generating a hash of the code a component and comparing the hash to a hash retrieved in a signed and trusted manifest or certificate associated with the component. In alternative exemplary embodiments, measurements performed by components of the ATM, TP and/or host banking system may correspond to any evaluation of a component which is operative to produce information useful for authenticating the component.

As discussed previously, an exemplary embodiment of the ATM may use a TP to securely store encrypted secrets on the ATM. Each trusted ATM component may access functions of the TP for use with encrypting (sealing) or decrypting (unsealing) secret data stored on the ATM. In addition exemplary embodiments of an ATM may be operative to store encrypted information on the ATM without using a TP. Such an ATM may include one or more secret keys for encrypting, decrypting, and other cryptographic process. Such keys must be available for use even after the machine has been shut down and restarted. Although such keys may be stored in a non-volatile media such as a hard drive, an exemplary embodiment may be operative to have periodic access to the secret keys without storing the keys on a potentially insecure hard drive or other long term storage medium. In this described exemplary embodiment, the ATM may be operative to generate and re-generate the secret keys each time the system is re-booted and/or when the secret keys are needed.

An exemplary embodiment of the ATM may generate or re-generate the identical key by combining keying material provided by the calling application, other software applications, the

-  
- operating system and/or hardware devices of the ATM. The combination of this keying material results in a key.

In this described exemplary embodiment the terminal control software for the ATM may include a function for generating the key. Portions of the keying material may be supplied as  
5 one or more arguments to the key generation function from a calling application. Such arguments may include for example a data name , an application name, a password, a machine identification number, or any other information that the calling application can reproducibly pass to the function. In an exemplary embodiment, the key generation function may require at least one of the arguments, while the other arguments may be optionally supplied by the calling  
10 application. For example, the data name argument may be required, whereas other arguments such as the application name, password, and machine identification number may be optional.

The data for these arguments may be compiled into the application that calls the key generation function. Also these arguments may be accessed by the calling application from outside the application and passed to the key generation function. For example, the calling  
15 application may pass a machine identification number to the function which is unique to the ATM. Such a machine identification number may correspond to a processor ID, an operating system machine ID, or any other information which is unique to the computer, hardware, or software of the ATM and which is accessible to the application calling the key generation function. In a further exemplary embodiment, the key generation function may be operative to  
20 directly access the machine identification number. In this case, the key generation function may accept a boolean argument from the calling application which specifies whether or not the key

generation function should access and use the machine identification number as part of the keying material.

In the exemplary embodiment, the key generation function may also use one or more constant or static data items for use as additional keying material. Such constant data may be defined within the key generation function.

The exemplary embodiment of the key generation function may use one or more hashing functions such as MD5 or SHA-1 to produce a one-way hash number from a combination of the keying material available to the key generation function. The resulting one-way hash number may correspond to a 128 bit value that is usable as a reproducible key for encrypting and decrypting information with the ATM.

To further increase the security of the encrypted data, the ATM may use the generated key in combination with a random salt value. Each time the generated key is used to encrypt data, the ATM software performing the encryption may generate a random or unique salt value. the salt value may be hashed with the generated key to produce a variable or salted key. The salted key may then be used to encrypt the data at the ATM with a symmetrical encryption algorithm such as DES for example. Because the salted key can only be re-produced if the salt value is known, the software which generated the encrypted data using the salted key may append the salt value to the encrypted data.

To decrypt the encrypted data, the salt value may be removed from the encrypted data. The ATM application decrypting the data may access or re-generated the generated key using the previously described key generation function. The generated key may then be hashed with

-  
- the salt value associated with the encrypted data to re-generate the salted key. The salted key may then be used to decrypt the encrypted data using the symmetrical algorithm

In addition to encrypting and decrypting ATM data using the generated and resulting salted key, the ATM may also perform other cryptographic operations using the generated and  
5 resulting salted keys such as generating a message authentication code (MAC) for ATM data.

Thus the new automated banking machine component authentication system and method achieves one or more of the above stated objectives, eliminates difficulties encountered in the use of prior devices and systems, solves problems and attains the desirable results described herein.

10 In the foregoing description certain terms have been used for brevity, clarity and understanding, however no unnecessary limitations are to be implied therefrom because such terms are used for descriptive purposes and are intended to be broadly construed. Moreover, the descriptions and illustrations herein are by way of examples and the invention is not limited to the exact details shown and described.

15 In the following claims any feature described as a means for performing a function shall be construed as encompassing any means known to those skilled in the art to be capable of performing the recited function, and shall not be limited to the features and structures shown herein or mere equivalents thereof. The description of the exemplary embodiment included in the Abstract included herewith shall not be deemed to limit the invention to features described  
20 therein.

Having described the features, discoveries and principles of the invention, the manner in which it is constructed and operated, and the advantages and useful results attained; the new and

useful structures, devices, elements, arrangements, parts, combinations, systems, equipment, operations, methods and relationships are set forth in the appended claims.